UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/078,933	05/14/98	BLANDY G	AT9-98-071

EXAMINER	
DAS,C	
ART UNIT	**PAPER NUMBER**
2762	3

LM02/0914

DUKE E YEE
P O BOX 802334
DALLAS TX 75380

DATE MAILED: 09/14/99

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary

Application No.	Applicant(s)
09/078,933	Blandy
Examiner	**Group Art Unit**
Chameli Das	2762

☒ Responsive to communication(s) filed on *May 14, 1998*.

☐ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire *3* month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claims

☒ Claim(s) *1-32* is/are pending in the application.

Of the above, claim(s) _____ is/are withdrawn from consideration.

☐ Claim(s) _____ is/are allowed.

☒ Claim(s) *1-32* is/are rejected.

☐ Claim(s) _____ is/are objected to.

☐ Claims _____ are subject to restriction or election requirement.

Application Papers

☒ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

☐ The proposed drawing correction, filed on _____ is ☐approved ☐disapproved.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) _____.

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____.

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of References Cited, PTO-892

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

☐ Interview Summary, PTO-413

☒ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

--- SEE OFFICE ACTION ON THE FOLLOWING PAGES ---

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-3 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wiecek, US Patent No. 5,210,837 in view of Liedberg, US Patent No. 5,943,687.

As per claim 1, Wiecek teaches **identifying a path** within the routine is shown in column 21 line 41-44 ("**identifying as a path member** selected ones of the nodes in the paths between the graph entry nodes and the graph exit nodes in the initial program flow representation"), **being executed** is shown in column 7 line 53-57 ("An arc or an edge from a first node to a second node exists if the second node is a possible destination (or fall-through path) from the first node, as reflected by the order in which the blocks of instructions in the source program **are executed**"), a plurality of first type instructions are associated with the path is shown in column 2 line 48-56 ("The computer program contains sequentially-ordered blocks of consecutive instructions written in a first computer language, and the initial program flow representation includes a plurality of nodes each representing a different one of the blocks of instructions. The nodes in the initial program flow representation are connected in paths reflective of the possible sequences which

the blocks of instructions may be executed"), translating the first type instructions for the path being executed, wherein first type instructions are translated into second type for execution is shown in column 26 line 10-18 ("converting the first program into an initial program flow representation including a plurality of nodes each representing a different one of the blocks of instructions, the nodes being connected in paths reflective of the possible sequences which the blocks of instructions in the program may be executed in accordance with the first computer program, and the connection in any of the paths").

Wiecek does not specifically teach instruction for **unexecuted** path remain untranslated. However, Liedberg teaches unexecuted path in column 7 line 58-62 ("one might perform a more sophisticated analysis of the contents of the instruction buffers in the re-order buffer in order to count only those **non-executed instructions**"), non executed instructions inherently including untranslated path as claimed.

It would have been obvious to one of the ordinary skill in the art at the time of invention was made to combine Weicek's translating method with Liedberg's non-executed method because one of the ordinary skill in the art would be motivated to increase the processing speed of the computer system.

As per claim 2, Wiecek teaches executing instructions for a path response to a loop is shown in column 4 line 57-60 ("Loops include ones of the paths that reflect sequences of repetitive instruction block execution, and the ones of the nodes lying in the paths in loops are deemed members of a loop").

As per claim 3, Wiecek teaches translated instruction for the path are executed in an order is shown in column 3 line 21-32 ("A data processing system according to this invention transforms an initial program flow representation of a computer program into a modified program flow representation. The computer program contains sequentially-ordered blocks of consecutive instructions written in a first computer language, and the initial program flow representation includes a plurality of nodes each representing a different one of the blocks of instructions. The nodes in the initial program flow representation are connected in paths reflective of the possible sequences which the blocks of instructions may be executed in accordance with the computer program"),

4. Claims 4- 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kolawa et al, US Patent No. 5,784,553 in view of Liedberg, US Patent No. 5,943,687.

As per claim 4, Kolawa teaches identifying a path within the method that is being executed is shown in ABSTRACT line 21-23 (" A dynamic symbolic execution consists of a symbolic execution of the program performed along the path that corresponds to the trial set of actual inputs"), bytecodes are associated with the pats is shown in column 21 line 18-21 (" The SVM symbolically interprets the JAVA program under test using information stored in the JAVA Bytecodes 210. Upon completion, symbolic expressions and associated path conditions are output by the SVM for all required branch conditions (block 302)"), compiling the byte codes is shown in ABSTRACT line 3-5 ("The JAVA program comprises program statements and program variables represented as JAVA source code and compiled by a JAVA compiler into

JAVA bytecodes"), byte codes are compiled into native machine code is shown in column 17 line 51-54 ("The bytecodes are a relatively high-level representation of the source code so that some optimization and machine code generation (via a just-in-time compiler 214) can be performed at that level"),

Kowala teaches bytecodes remain uncompiled is shown in column 5 line 53-55 (" The original source code 11 comprises all types of files used to express an uncompiled, that is, non-object code, computer program, including definitional and declarative files").

Kowala does not specifically teach bytecodes for unexecuted paths remain uncompiled. However, Liedberg teaches unexecuted path in column 7 line 58-62 ("one might perform a more sophisticated analysis of the contents of the instruction buffers in the re-order buffer in order to count only those **non-executed instructions**"), non executed instructions inherently including untranslated path as claimed.

It would have been obvious to one of the ordinary skill in the art at the time of invention was made to combine Kowala's byte code method with Liedberg's non-executed method because one of the ordinary skill in the art would be motivated to continue compilation of unexecuted paths when these paths are executed for the efficient optimization of the computer system.

Claim 5 is rejected under the same reason set forth in connection of rejection of claim 2.

Claim 6 is rejected under the same reason set forth in connection of rejection of claim 3.

As per claim 7, Kolawa et al teach JIT in compiling method is shown in column 17 line 62-65 ("or optionally turned into machine code by the Just-In-Time Compiler 214. The JAVA

Interpreter and Just-In-Time Compiler operate in the context of a Runtime System 222")

.As per claim 8, Kolawa et al teach data structure as claimed is shown in column 6 line 21-23 (" FIG. 2B is a data structure for storing a parse tree which describes the syntactic structure of the original computer program").

As per claim 10, data structure is a JIT station is shown in column 5 line 52-54 ("FIG. 2D is a data structure for storing block and branch analysis information extracted from the original computer program"), storing block and branch analysis information inherently including JIT compile information as claimed.

As per claim 11, identifying the method that is to be executed is shown in column 20 line 36-38 ("Module 41 identifies the branch condition controlling the execution of that code block"), compiling the bytecodes is shown in column 26 line 44-47 ("FIG. 23 is a flow diagram of the steps for handling JAVA bytecode instrumentation. At step 492, the JAVA Parser 230 reads the .class files 210 generated by the JAVA Compiler 208 and the .java files 200 of the original source code").

As per claim 12, Kowala teaches bytecodes remain uncompiled is shown in column 5 line 53-55 (" The original source code 11 comprises all types of files used to express an uncompiled, that is, non-object code, computer program, including definitional and declarative files").

Kowala does not specifically teach bytecodes for unexecuted paths remain uncompiled.

However, Liedberg teaches unexecuted path in column 7 line 58-62 ("one might perform a more sophisticated analysis of the contents of the instruction buffers in the re-order buffer in order to

count only those **non-executed instructions**"), non executed instructions inherently including untranslated path as claimed.

It would have been obvious to one of the ordinary skill in the art at the time of invention was made to combine Kowala's byte code method with Liedberg's non-executed method because one of the ordinary skill in the art would be motivated to continue compilation of unexecuted paths when these paths are executed for the efficient optimization of the computer system.

Claim 13 is rejected under the same reason set forth in connection of the rejection of claim 3.

Claim 14 is rejected under the same reason set forth in connection of the rejection of claim 2.

Claim 15 is rejected under the same reason set forth in connection of the rejection of claim 8.

As per claim 16, Kolawa teach data structure stores the instruction is shown in column 4 line 36-38 ("IG. 2D is a data structure for block and branch analysis information stored in the program database of FIG. 2A")

Claim 17 is rejected under the same reason set forth in connection of the rejection of claim 11.

Claim 18 is rejected under the same reason set forth in connection of the rejection of claim 2.

Claim 19 is rejected under the same reason set forth in connection of the rejection of claim 3.

Claim 20 is rejected under the same reason set forth in connection of the rejection of claim 10.

Claim 21 is rejected under the same reason set forth in connection of the rejection of claim 8.

Claim 22 is rejected under the same reason set forth in connection of the rejection of claim 9.

Claim 23 is rejected under the same reason set forth in connection of the rejection of claim 10.

Claim 24 is rejected under the same reason set forth in connection of the rejection of claim 11.

Claim 25 is rejected under the same reason set forth in connection of the rejection of claim 12.

Claim 26 is rejected under the same reason set forth in connection of the rejection of claim 3.

Claim 28 is rejected under the same reason set forth in connection of the rejection of claim 8.

Claim 29 is rejected under the same reason set forth in connection of the rejection of claim 16.

Claim 30 is rejected under the same reason set forth in connection of the rejection of claim 11.

Claim 31 is rejected under the same reason set forth in connection of the rejection of claim 3.

Claim 32 is rejected under the same reason set forth in connection of the rejection of claim 2.

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Breternitz, Jr. et al teach method and apparatus for sequencing computer instruction execution in a data processing system, US Patent No. 5,889,999.

Levine et al teach system and method for performance monitoring of instructions in a re-order buffer, , US Patent No. 5,938,760.

Hamby et al teach incremental byte code compilation system, US Patent No. 5,848,277.

Hansen et al teach customer programmable real-time system US Patent No. 4,747,127.

Hansen et al teach control of real-time systems utilizing a nonprocedural language, US Patent No. 4,695,977.

AN: 97: 278366, Title: Java Battleground: Virtual Machine, Author: Moeller, Michael, source: PC week, 19th May, 1997.

AN: 98: 522515, Title: Java compilers take diverse paths to market, Author: Cole, Bernard, source: Electronic Engineering Times, 12th October, 1998).

AN: 96:89491, Title: Small Talk solutions '96 Conference, Source: Computer International, 15 March, 1996.

AN : 406005, Title: Sun Reveals First Java Processor Core, source: Microprocessor Report, 28th October, 1996.

AN: 9725787, Title: Java Processors balance power and cost, Source: Electronic Engineering Times, (13th Jan 1997).

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chameli Das whose telephone number is 703-306-3014. The examiner can normally be reached on Monday-Friday from 8:00 A.M to 4:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor Tariq Hafiz can be reached at 703-305-9643. The fax number for this group is 703-308-1396. An inquiry of general nature or relating to the status of this application or proceeding should be directed to the group receptionist whose telephone number is 703-305-9600.

CDAS

9/8/99

Tariq R. Hafiz
Supervisory Patent Examiner
Technology Center 2700